# *ECE-205 Lab 9*

*Frequency Response and Experimental Bode Plot Construction*

*Overview*

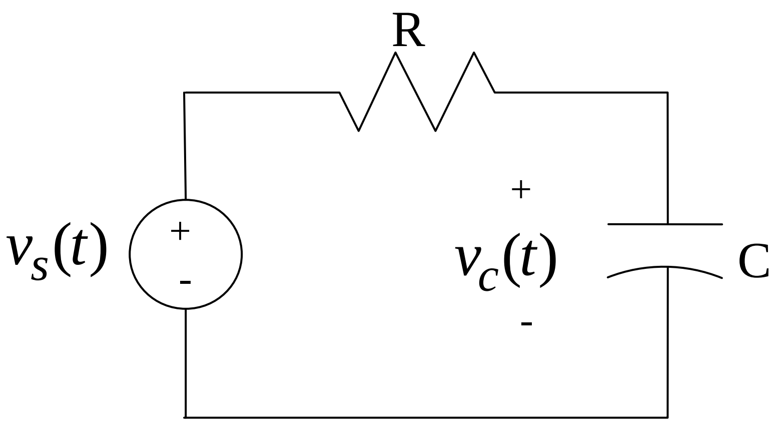
*In this lab you will build and measure the frequency response of two circuits and then use Matlab to construct a Bode plot of each system. For many unknown (or complex) systems this is a very common method for determining the transfer function. You will also need to modify a somewhat complex Matlab function. You only need to make a few changes, but often one learns to program by modifying code that already does something close to what you want it to do. You will need to download two Matlab files from the class website.*

## PART I : Frequency Response of a First Order System

**1)** Using a bread board construct the following circuit using a 1 $\mu$ f capacitor and a 1 k$\Omega$ resistor. These components should be available in your lab kit. We will measure the output of the circuit as the voltage across the capacitor. For this simple circuit the transfer function is of the form

$$H(s) = \frac{K}{\tau s + 1}$$

where $K$ is the static gain (it is one for this model), and $\tau$ is the time constant (and is equal to $RC$ in this case).



For this circuit we then expect the time constant to be around 1 millisecond, $\tau = 0.001$. For this first order system we then expect the corner frequency to be at $\omega = 2\pi f = \frac{1}{\tau}$, or $f = \frac{1}{2\pi\tau} = \frac{1}{2\pi(0.001)} \approx 159Hz$.

**2)** Connect the Mobil Studio board to your circuit, using the signal generator as the input voltage. The input will initially be a 20 Hz sine wave with a 2.0 volt peak to peak voltage (an *amplitude* of 1.0 volt) with 0 DC offset. Set the coupling to DC coupling.

Measure the input voltage as the difference between A1+ and A1- (never trust a signal generator!), and measure the output voltage as the difference between A2+ and A2-. You will need to use the cursors for this lab, so be sure to turn them on. Set the time scale so you see one or two periods of the input and output sine waves.

**3)** For an input $x(t) = A\cos(\omega_o t)$ we know the steady state output wil be

$$y_{ss}(t) = A|H(j\omega_o))|\cos(\omega_o t + \angle H(j\omega_o))$$

Hence if we measure the input and output amplitude at each input frequency $\omega_o$, it should be pretty easy to determine $|H(j\omega_o)|$ at each of these frequencies. To determine the phase, we will look at the time delay between the input and output signals at each input frequency, as follows,

$$\cos(\omega_o(t-t_d)) = \cos(\omega_o t - \omega_o t_d) = \cos(\omega_o t + \angle H(j\omega_o))$$

Note that in this relationship the phase of the transfer function is measured in radians, $\angle H(j\omega_o) = -\omega_o t_d$

We can then convert the phase from radians to degrees using

$$\angle H(j\omega_o) \text{ (in degrees)} = -2\pi f_o\, t_d \times \left[\frac{180^o}{\pi}\right] = -360^o\, f_o \times t_d$$

Here is our (initial) procedure then:

- Choose a frequency for the input signal (keep the input amplitude at 1 volt, or 2 volts peak to peak)
- Using the cursors, measure the input amplitude (not peak to peak)
- Using the cursors, measure the output amplitude (not peak to peak)
- Using the cursors, measure the time of the peak of the input signal
- Using the cursors, measure the time of the next peak of the output signal. This must be the peak immediately following the peak of the input signal since we are measuring the time delay between these signals.

**3)** Rather than recording this and then typing it into Matlab, we will record our data directly into the Matlab program **process_data_a.m.** You should open this file and fill in the matrix **measured** as you go along. Each row in this matrix contains the data for a specific input frequency. The columns contain the data

- Frequency (in Hz)
- Amplitude of the input signal (in mV)
- Amplitude of the output signal (in mV)
- Time of the peak of the input signal (in milliseconds)
- Time of the subsequent output peak (in milliseconds)

Put a space between each entry, and end each row (except the last row) with a semicolon (;). Note that this function modifies the contents of each column to produce the data we want to use.

**4)** For your initial attempt (you will add more points later), collect the appropriate data for input frequencies 20, 60, 100, 140, 180, 220, 280, 320, 520, 720, 920, and 1120 Hz.

**5)** Save the program **process_data_a.m.** This program returns an array to the calling program. In the Matlab window, type

**data = process_data_a;**

**6)** The program **model_a.m** uses this data, as well as knowledge of the expected transfer function, to best fit the transfer function to the measured data and produce a couple of nice plots. There are three input arguments to this program, the array **data**, the estimated **gain**, and the estimated **time constant**. How well this program works in part depends on how well you can guess these values. If we assume our gain is 1and our time constant is 0.001, to run the program at the Matlab command window we type

**model_a( data, 1.0, 0.001)**

Matlab will spit a bunch of stuff to the screen, and then eventually you will get magnitude and phase plots. If the plots look way off, one of two things has happened:

- Your initial guess was way off
- Your data, or at least some of it, was junk

At this point you may want to check any data that does not look too good and rerun the programs before you go on. Remember that you need to rerun **process_data_a** if you change your data.

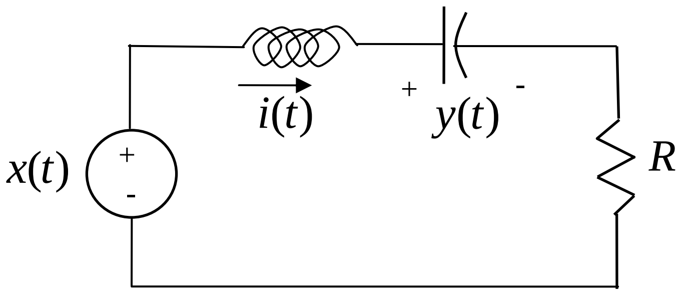
**7)** Add more data points to fill out your graph (spread throughout the frequency range). Your final plot should have at least 20 frequencies. *Note that you can just add your new data to the end of the matrix* ***measured*** *since it gets sorted.*

*Include your final Bode plot figure in your memo. You will also have to attach your file* ***process_data_a.m*** *in your e-mail.*

**PART II : Frequency Response of a Second Order System**

In this part we do basically the same thing as in the first part, but you have to change some of the Matlab files.

**1)** Construct the following second order circuit:



For this circuit, use your **variable resistor** for R, use a 0.01 $\mu$f capacitor, and a 33 mH inductor. For this system the transfer function is

$$H(s) = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

where $K$ is the static gain, $\zeta$ is the damping ratio, and $\omega_n$ is the natural frequency.

**2)** Set up Mobile Studio to provide the input and measure the output. The input should be a 1.0 kHz square wave with a 0.0 volt offset and 2 volts peak to peak. The output is the voltage across the capacitor.

**3)** Adjust the variable resistor so there is a 100% overshoot for a step response. *Include a scope capture of this step response in your memo.* Do not touch the resistor again.

**4)** Copy the file **process_data_a.m** to a new file **process_data_b.m.** You will be modifying this new file as you collect data for this system.

**5)** Record the necessary data for frequencies 1000, 3000, 7000, 8500, 9000, 9500, 11000, 13000, and 15000 Hz.

**6)** Generate the data array by typing **data = process_data_b** in the Matlab command window.
If your do not put a semicolon after this command, the data array will be printed out. The first column of this array is the frequency in Hz, and the second is the gain. The frequency with the highest gain is the *resonant* frequency, and you should use this as your first estimate of the natural frequency. However, you will need to convert this frequency to radians/second.

**7)** Copy the file **model_a.m** to the file **model_b.m.** You are now going to have to modify this new file to find the optimal fit to the ***new*** transfer function. Specifically, your end result must do/have the following:

- There must be four input arguments to the program, the data array, the static gain, the damping ratio, and the natural frequency. *These must be input arguments to the function or you will lose points.*

- Change the comments in the code to match what the code is doing. Pay attention in particular to the comments about the input arguments.

- Your final plot must have a caption that includes the gain, the damping ratio, and the natural frequency. Note that to get Matlab to generate these last two in a nice font, type \zeta and \omega_n.

- Your must have a good fit between your measured data and your estimated transfer function.

- Note that if any of your parameters are negative, you need to take the absolute values of the parameters when you extract them from the x array. *You cannot have any negative parameters*

**8)** Record more data points to fill out your Bode plot estimate, particularly near the resonant peak. You should end up with at least 20 data points spread out among the frequencies.

*Include your final graph in your memo. Attach your m-files* ***process_data_b.m*** *and* ***model_b.m*** *to your e-mail*

## PART III: So What Does a Bode Plot Tell You?

Students often think of Bode plots as just and easy way to get easy points on an exam by following a bunch of rules, and completely forget the information you get from the Bode plot. Having just constructed Bode plots yourself, this should be easy…..

For system A, whose frequency response is displayed in the Bode plot in Figure 1, estimate the steady state output if the input to the system is

$a)\ x(t) = 3\cos(10t)$

$b)\ x(t) = 5\sin(40t + 50^o)$

For system B, whose frequency response is displayed in the Bode plot in Figure 2, estimate the steady state output if the input to the system is

$c)\ x(t) = 50\sin(10t)$

$d)\ x(t) = 100\cos(200t + 35^o)$

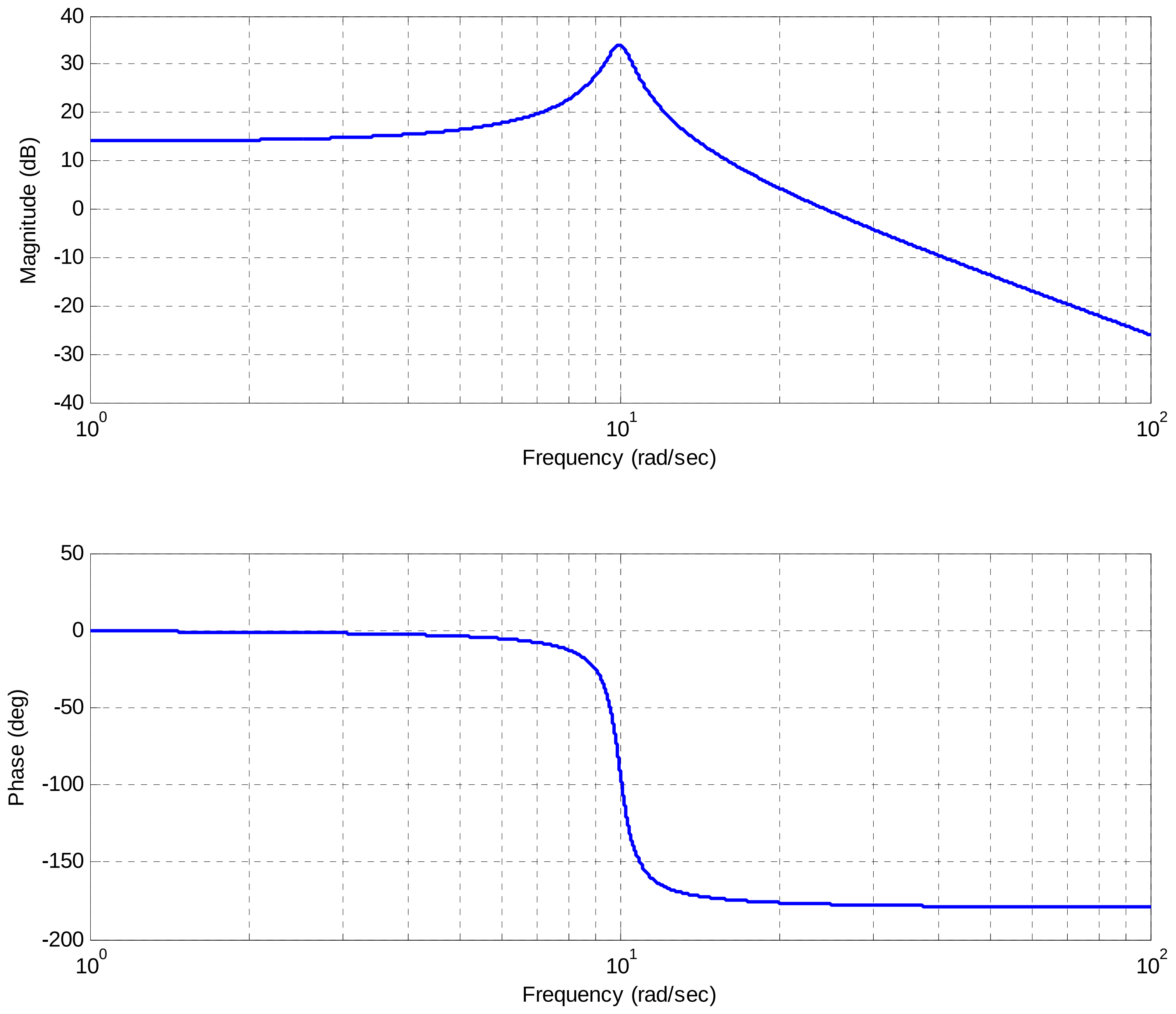


**Figure 1: Frequency response of system A**

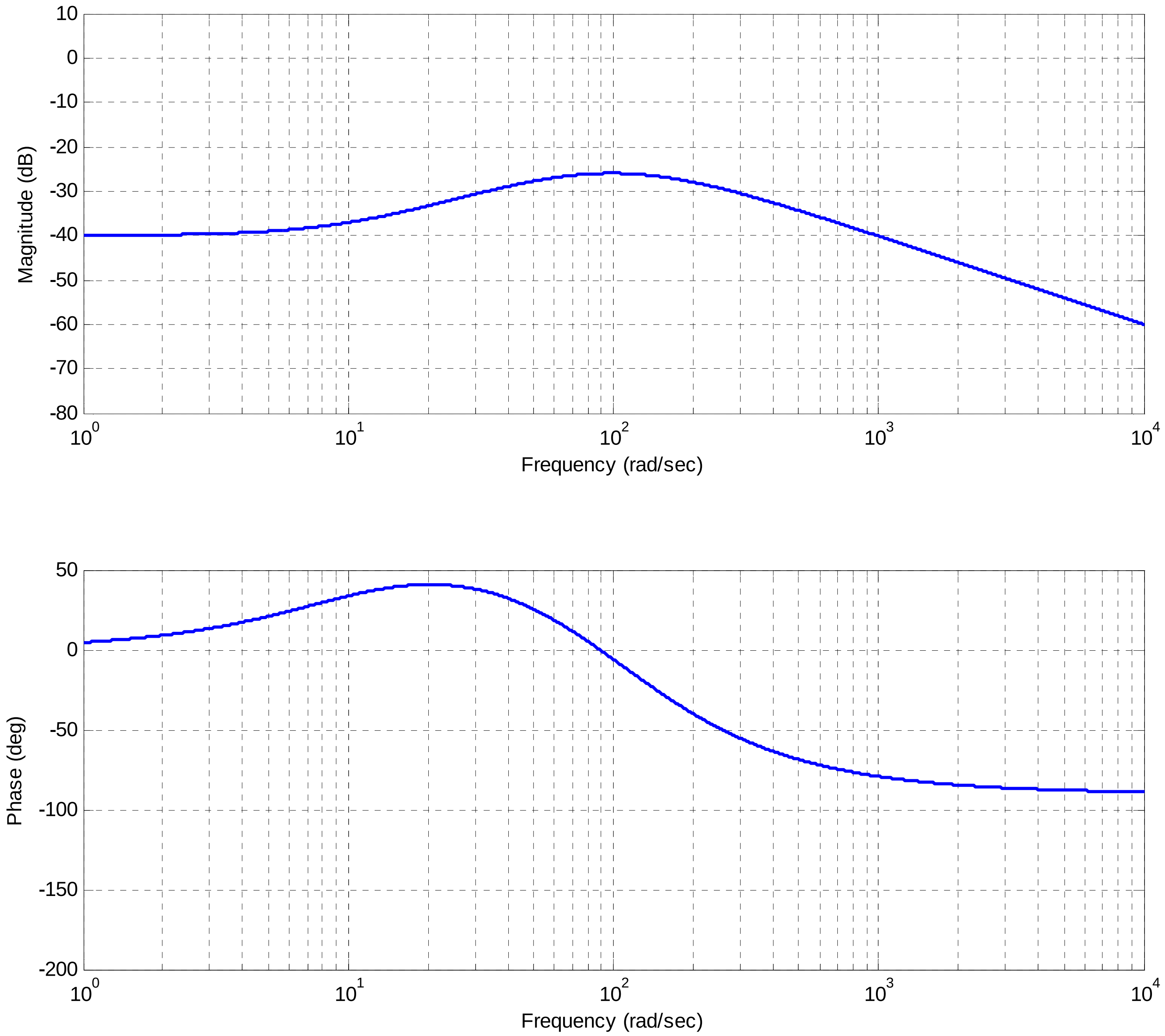


**Figure 2: Frequency response of system B**